



北京理工大学
BEIJING INSTITUTE OF TECHNOLOGY

软件理论与工程

Software Theory and Engineering

Dr. Guangyu (Ryan) Gao
Associate Professor, School of Software
Beijing Institute of Technology

Email: guangyu.ryan@yahoo.com
<http://guangyugao.weebly.com>

第1章 软件工程概述

讲授内容

- 软件工程的基本概念
- 基于计算机的系统工程
- 软件过程



1. 什么是软件？

- 软件（英语：software）是一系列按照特定顺序组织的电脑数据和指令，是电脑中的非有形部分。
- 软件包括所有在电脑运行的程序，和其架构无关，例如可执行文件、库及脚本语言都属于软件。
- 软件是以抽象思维的方式对现实世界的高度概括和抽象。软件是定义计算的逻辑制品，其实质是以计算为核心手段实现应用目标的解决方案。
- 软件并不一定只包括可以在计算机上运行的计算机程序，与计算机程序相关的文档，一般也被认为是软件的一部分。

梅宏院士：软件一定是未来世界非常重要的一个元素，无处不在的软件、软件定义的未来世界。

1. 什么是软件？

- Software=program+data+document
- 软件分类：
 - 系统软件，应用软件，中间件等。
 - 通用软件、专用软件等。
- 软件演化发展类似于社会的发展，是不断分工协作，高度细化的过程。
- 不断抽取软件共性，即抽取出了程序的共性（稳定）成分，例如从“硬件+程序”到“硬件+操作系统+应用程序”。

2. 什么是软件工程？

- 1968年秋季，NATO（北约）的科技委员会召集了近50名一流的编程人员、计算机科学家和工业界巨头，讨论和制定摆脱“软件危机”的对策。Fritz Bauer在会议上首次提出“软件工程”概念。
- The establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines.” -- Fritz Bauer, 1968



2. 什么是软件工程？

- 软件危机是指落后的软件生产方式无法满足迅速增长的计算机软件需求，从而导致软件开发与维护过程中出现一系列严重问题的现象。
 - IBM OS/360操作系统被认为是一个典型的案例。OS/360系统的负责人Brooks在随后他的大作《人月神话》中曾经承认，在他管理这个项目的时候，他犯了一个价值数百万美元的错误。
 - 1963年美国飞往火星的火箭爆炸，造成1000万美元的损失。原因是FORTRAN程序 DO 5 I=1, 3，误写为：DO 5 I=1.3。
 - 1967年苏联“联盟一号”载人宇宙飞船在返航时，由于软件忽略一个小数点，在进入大气层时因打不开降落伞而烧毁。
 - 千年虫问题，或2000年问题，计算机在处理2000年1月1日以后的日期和时间时，可能会出现不正确的操作，从而可能导致一些敏感的工业部门（比如电力，能源）和银行，政府等部门在2000年1月1日零点工作停顿甚至是发生灾难性的结果。



2. 什么是软件工程？

- 软件危机（英语：Software Crisis）的本源是复杂、期望和改变，是急剧增加的计算机力量带来的冲击和可能要处理的问题的复杂性的冲突。从本质上来说，它谈到了写出正确、可理解、可验证的计算机程序的困难。
- 软件危机背后的哲学思考：
 - 量变到质变：软件危机使人们认识到中大型软件系统与小型软件有着本质性差异：大型软件系统开发周期长、费用昂贵、软件质量难以保证、生产率低，它们的复杂性已远超出人脑能直接控制的程度，大型软件系统不能沿袭工作室的开发方式，就像制造小木船的方法不能生产航空母舰一样。
 - 不以规矩，不能成方圆：以系统工程的思维开展软件设计，积极运用面向对象技术等。

软件危机的解决途径

- (1) 吸取经验：推广使用在实践中总结出来的开发软件的成功技术和方法
- (2) 创新探索：研究探索更有效的技术和方法
- (3) 善假于物：开发和使用更好的软件工具
- (4) 工程思维：认识到软件是一种组织良好、管理严密、协同配合的工程项目

2. 什么是软件工程 ?

■ IEEE: 软件工程

(1) 将系统化的、规范的、可度量的方法应用于软件的开发、运行和维护的过程，即将工程化思想应用于软件开发过程中

(2) 上述方法的研究

■ 软件工程的目标：低成本，高质量，按时交付

⚙️ 3. 软件工程的本质特性

- 关注大型程序的构造
- 如何控制复杂性
- 软件需求不断变化
- 旨在提高软件开发的效率
- 团队合作是软件工程顺利实施的关键
- 软件必须有效支持它的用户
- 由一种文化背景的人替另一种文化背景的人创造产品（设计者、使用者）

4. 软件工程所面临的主要问题

- 遗留系统的挑战 (Legacy systems)
- 交付上的挑战
- 复杂性与大量的细节 (火星探测器的失败)
- 技术的不确定性 (技术的发展与开发人员对技术的理解程度都不同)
- 由于交流障碍而引起的需求不确定性
- 需求是持续变化的
- 不断的修改所带来的错误使得软件退化
- 人为和市场的风险
- 软件费用、可靠性、生产率、重用问题难以解决



5. 加快建设世界科技强国

- 软件在支持这些传统领域产业结构升级换代甚至颠覆式创新的过程中起到核心关键作用，加速重构了全球分工体系和竞争格局。高效地构建和运用高质量软件系统的能力成为国家和社会发展的一种核心竞争力。

作为新一轮科技革命和产业革命的标志，德国的“工业 4.0”和美国的“工业互联网”均将软件技术作为发展重点。

《国务院关于印发鼓励软件产业和集成电路产业发展若干政策的通知》（国发〔2000〕18号）

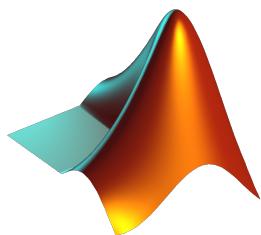
《国务院关于印发进一步鼓励软件产业和集成电路产业发展若干政策的通知》（国发〔2011〕4号）

《国务院关于印发新时期促进集成电路产业和软件产业高质量发展若干政策的通知》（国发〔2020〕8号）

5. 加快建设世界科技强国

卡脖子技术之软件行业

- 微软和谷歌形成了对全球PC、移动终端的操作系统垄断。
- 目前华为的EDA软件已经被断供，哈工大学生无法使用MATLAB就是其中的例证。
- 电子制图所需的CAD、芯片设计所需的EDA、工业制动画控制系统MEMS、工业设计和仿真软件MATLAB等软件产品， C/C++、Python、Java、PHP、CS、GO等编程语言以及各种软件标准、行业标准均是西方国家所定义，随时有被断供风险。



Mentor
Graphics®
SIEMENS
cadence
synopsys®



5. 加快建设世界科技强国

- 科技创新是中华民族实现伟大复兴的助推器。
- 在创新驱动的战略下，逐步迈向“中国创造”和“中国智造”。
- 党的十九大报告中，10余次提到科技，50余次强调创新。
- 而在今年全国两会期间，“创新”和“科技”再次成为两大热议的关键词。
- 在党的创新驱动发展战略的引领之下，五年来，我国创新驱动发展成果丰硕。我国科技创新由跟跑为主转向更多领域并跑、领跑，成为全球瞩目的创新创业热土。

3. SW and CS

- 计算机科学研究的是构成计算机和软件系统基础的有关理论和方法，而软件工程则研究软件制作中的实际问题。
- 理论上，所有软件工程都应该以计算机科学理论作为坚实的基础，但对付实际的、复杂的问题时需要用软件工程的方法来解决。

4. SW and Systems Engineering

- 系统工程的产生比软件工程早。
- 基于计算机的系统工程，研究由软件起主导作用的、有关负责系统的开发和进化的方方面面，包括硬件开发、系统决策、过程设计、系统实施和软件过程等。

7. 软件工程的成本及成本分布

- 软件成本分布取决于所采用的软件过程和所开发的软件类型。
- 一般的成本分布：

描述:设计:开发:集成与测试 (15:25:20:40)

- 需求极高的软件系统的集成与测试成本50%
- 软件进化的成本
- 软件维护的成本
- 基于Web的电子商务系统的成本模型

8. 软件工程方法 (Method)

- 软件工程方法是软件工程学科核心内容，不同软件开发方法有其各自特征，典型的软件开发方法
 - 结构化方法
 - 模块化方法：
 - 面向数据结构的软件开发方法
 - 面向对象方法
 - 基于构件的软件开发方法

UML (Unified Modeling Language)

- 1994，OO思想已经贯穿整个软件生存期，具有影响的OOA&D方法达50余种。
- UML是一种对软件密集型系统进行可视化、详述、构造和文档化的建模语言，主要用于分析和设计阶段的系统建模。

10. 优良软件的属性

- 功能属性 (Functional)
- 非功能属性 (Non-Functional)

11. 软件工程所面临的主要问题

- 遗留系统的挑战 (Legacy systems)
- 交付上的挑战
- 复杂性与大量的细节（火星探测器的失败）
- 技术的不确定性（技术的发展与开发人员对技术的理解程度都不同）
- 由于交流障碍而引起的需求不确定性
- 需求是持续变化的
- 不断的修改所带来的错误使得软件退化
- 人为和市场的风险
- 软件费用、可靠性、生产率、重用问题难以解决

讲授内容

- 软件工程的基本概念
- 基于计算机的系统工程
- 软件过程

1. 系统总体特性

- 功能特性
- 非功能特性，如可靠性、安全性、保密性等。
- 与系统总体可靠性紧密关联的3个方面：
 - 硬件可靠性
 - 软件可靠性
 - 操作可靠性

2. 系统及其环境

- 系统都是在一定的环境中存在的。
- 系统环境必将影响系统的功能和性能。
- 环境包含一系列相互作用的其它系统，有时，环境可能被看作是一个独立的系统。
- 系统工程师一定要了解系统环境
 - 在许多情况下，系统的目的就是要改变环境。
 - 一个系统的功能要受到环境变化的影响，这种影响可能很难估计。

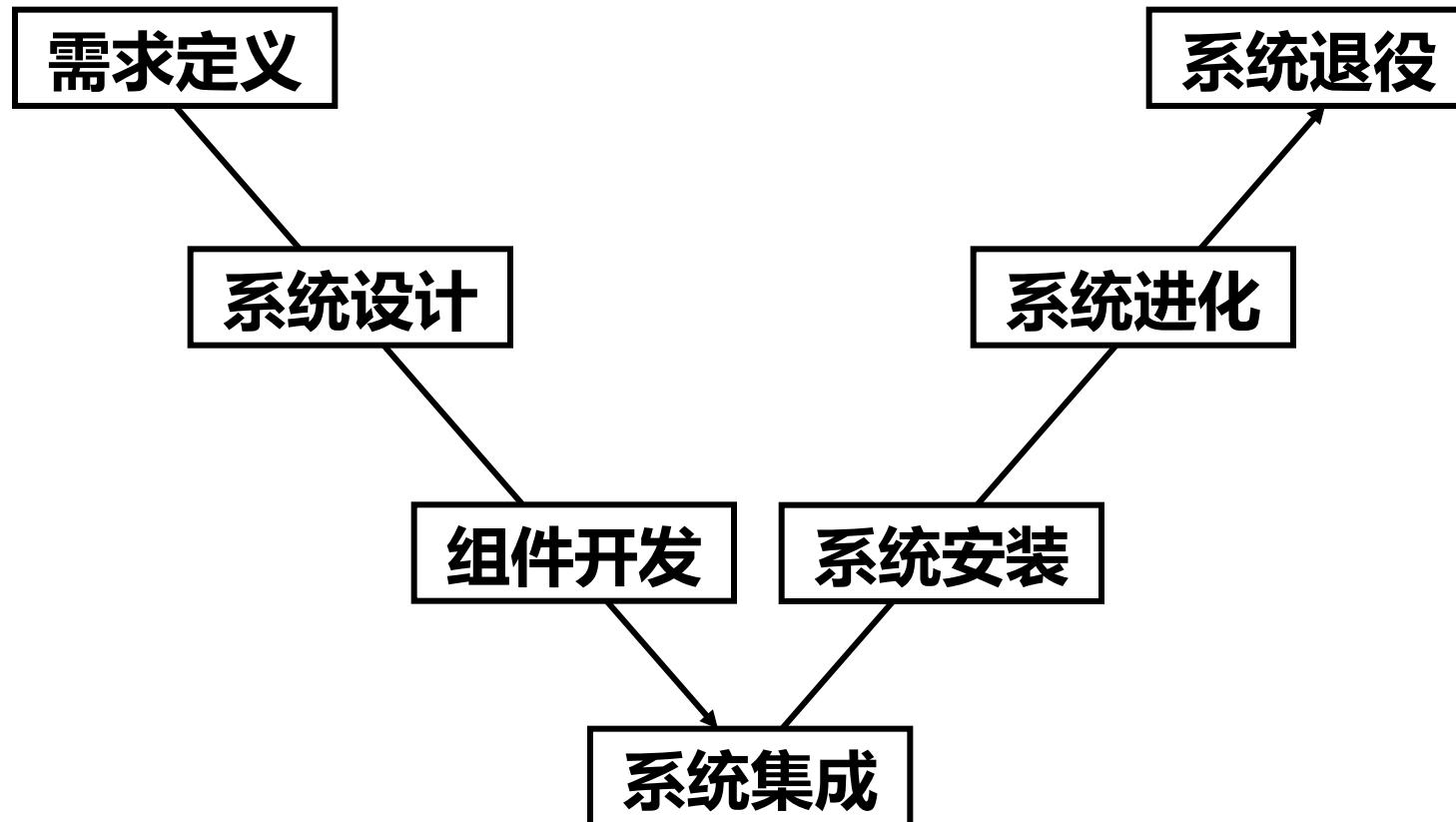
3. 系统建模

- 系统建模作为系统需求和设计活动的一部分，描述组成系统的组件以及组件之间的关系（系统体系结构）
- 系统体系结构通常以方块图来描述。
- 一个系统体系结构可以以功能为单位划分成子系统或功能组件，而不必关心由硬件还是软件实现。

■ 功能组件可以分为六类：

- 传感器组件：收集来自系统环境的信息。
- 执行机构组件：引起一些系统环境的变化。
- 计算组件：给定输入，执行计算并产生输出。
- 通信组件：实现系统组件之间的彼此通信。
- 调度组件：协调组件间的操作。
- 接口组件：将一个组件中的表示转换成另一个组件中的表示。

4. 系统工程过程



■ 系统需求定义注重三种类型的需求：

- 抽象的功能需求
- 系统特性
- 系统一定不能具有的性质

■ 系统设计所包括的活动有：

- 分割需求（通常有多种分割方法，可以产生不同的方案）
- 识别子系统
- 为子系统分配需求（通常由于外购COTS子系统带来的限制而需要修改需求）
- 描述子系统的功能
- 定义子系统的接口

■ 子系统开发所常见的问题:

- 通常可以并行开发不同的子系统
- 一些超出子系统范围的修改经常发生，但是对已经实现的硬件工程的修改往往带来非常昂贵的代价，这时修改往往落在软件上（软件的固有柔性，*flexible*），说明软件需求的变化更大更频繁。

■ 系统集成最好采取增量式过程，其理由是：

- 子系统的开发时间通常无法预计
- 增量式集成可以减少错误定位的成本（错误定位可能引起不同子系统的承包商之间的争论，解决问题的谈判须花费时日）

讲授内容

- 软件工程的基本概念
- 基于计算机的系统工程
- 软件过程

1. 什么是软件过程？

- 软件过程是指开发软件产品的一组活动及其结果。
- 所有的软件过程都包含4项基本的活动：软件描述、软件开发、软件有效性验证和软件进化。
- 不同的软件过程以不同的方式组织这4项活动，活动的结果会影响活动的进度。
- 不同的机构可能用不同的过程来制作同一类产品。

2. 为什么要定义软件过程？

- 一个过程定义了为达到确定的目标，需要什么人在什么时间以何种方式做何种工作
(Goal,Who,When,How,What)
- 对于 Customer, User, Developer, Manager，一个广泛适用的过程使得所有涉众更好地理解自己所扮演的角色、更清楚地知道自己及他人在什么时间做什么。
- 促使过程的有机结合和改善，以获得“最好过程”
- 可以使公司内部的培训标准化。
- 由于过程的可重复性，利于开发进度的安排，利于成本估算。

3. 什么是软件过程模型？

- 模型与建模
- 软件过程模型是从一特定的角度对软件过程的本质描述。
- 软件过程模型包括构成软件过程的各种活动、软件产品以及所有涉众（stakeholder）

3. 什么是软件过程模型？

- 软件开发过程(software development process), 或软件过程(software process), 是软件开发的开发生命周期(software development life cycle).
- 其各个阶段实现了软件的需求定义与分析、设计、实现、测试、交付和维护。软件过程是在开发与构建系统时应遵循的步骤，是软件开发的路线图。

3. 什么是软件过程模型？

- 一个软件过程模型是软件过程的一个抽象表示。
- 一般使用过程模型区分或解释不同的软件开发方法。在实际的软件开发中，很少单独使用单一的过程模型，
- 4类软件过程模型
 - 生命周期模型（瀑布模型）
 - 进化式开发模型
 - 形式化系统开发模型
 - 面向复用的开发模型

通用模型或范型（Paradigm, Methodology）

- Waterfall Model
- Waterfall Model with Maintenance Circle
- Waterfall Model with Prototyping
- Spiral Model（瀑布模型 + 原型模型）
- V Model
- Phased Development Model
- Incremental and Iterative Model
- RUP

2. 软件描述

- 可行性研究
- 需求导出和分析
- 需求描述
- 需求有效性验证

3. 软件设计与实现

- 体系结构设计
- 子系统的抽象描述（如包图）
- 接口设计
- 组件设计
- 数据结构设计
- 算法设计

4. 软件有效性验证

- 单元测试
- 功能模块测试
- 子系统测试
- 系统测试
- 验收测试（如 α 、 β 、 λ ）

5. 软件进化

- 软件进化，即软件在其生命周期内不断地随着需求的变更而变更的过程。
- 软件系统的柔性决定了软件进化的现实
 - 避免昂贵的硬件修改
 - 软件进化亦存在一些问题
- 软件进化可以看作是软件开发过程和软件维护过程的统一。